Designed for Windows Mobile™ Software Application Handbook for Smartphone

August 2003

Contents

August 2003

WHAT DOES THE LOGO MEAN? 3

SUMMARY OF UPDATES SINCE MAY 2003 VERSION 4

IMPORTANT NOTES ON INTERNATIONAL AND LOCALIZED VERSIONS 4

IMPORTANT NOTES ON APPLICATION TYPES 4

GENERAL REQUIREMENTS FOR SMARTPHONE APPLICATIONS 6

INSTALLATION/UN-INSTALLATION REQUIREMENTS 6

UI/SHELL SUPPORT REQUIREMENTS 9

FUNCTIONALITY REQUIREMENTS 10

SPECIAL CIRCUMSTANCES AND ADDITIONAL REQUIREMENTS 13

SDI/FILE-BASED APPLICATIONS 13

UTILITIES 13

ADD-ONS 13

DEVICE DRIVERS 13

HARDWARE 14

OTHER MICROSOFT LOGO PROGRAMS 15

WHAT DOES THE LOGO MEAN?

The Designed for Windows Mobile™ logo program was developed by Microsoft to help end-users easily identify software products that are compatible with Windows Mobile-based Smartphones. Smartphone users are assured that software products with the Designed for Windows Mobile logo are designed specifically for Smartphones, and incorporate new functionality featured in Windows Mobile 2002-based Smartphone where applicable. As an Independent Software Vendor (ISV), licensing the logo opens up a number of marketing opportunities.

- **Microsoft Mobile2Market**: Once your Windows Mobile-based Smartphone applications have been logo-certified, you can submit them to Mobile2Market, a program designed to help ISVs create new revenue opportunities for their applications by connecting them with mobile operators and retailers. Visit www.microsoft.com/windowsmobile/mobile2market.

- **Market Differentiation**: Once your applications have been logo-certified, you can use the logo on product and marketing materials to show that your product is compatible and designed for Windows Mobile 2002-based Smartphones .

- **Microsoft Certified Partner Program**: ISVs with one or more products that pass certification testing for this logo program are eligible to join the Microsoft Certified Partner Program at the Member level. For more information, visit www.microsoft.com/partner/partnering/certified/.

Software applications are tested by NSTL (http://www.nstl.com/logoprogram/win_ce_logoprograms.html), QualityLogic (http://www.qualitylogic.com/certification_programs.html), VeriTest (http://www.veritest.com/certification/ms/sphone/default.asp?visitor)and Wipro (http://www.wipro.com/prodesign/focusareas/embeddedsys/testingprograms.htm). For complete information about the logo testing process, including instructions on getting the necessary materials to get started as well as pricing, please visit their Web sites.

The Designed for Windows Mobile logo indicates that a software product provides all the features outlined in these guidelines. It is not a “quality assurance” seal. Neither Microsoft nor the independent test labs (NSTL, QualityLogic, VeriTest and Wipro) will test the quality of your product or ensure that it is “bug-free” as part of the certification program — certification testing is solely to make sure that your application performs all claimed functionality, that the logo requirements are met, and that your product is not generating frequent faults or system crashes.

Please note that the logo license agreement states: “You may only use the logo as a symbol that your Product has passed the applicable Designed for Windows Mobile-based Smartphone compatibility testing. You may not explicitly state or imply that Microsoft or the testing organization in any way endorses your product. Also, the logo program is not intended to be a “certification” program, that is, the logo does not represent that Microsoft or the testing organization certifies or warrants your product(s) in any way.”

Summary of Updates Since May 2003 Version

Removed:

- Microsoft MAPI for the Pocket PC Functionality Not Duplicated
- Dependency on Microsoft msgstore API
- No Dependency on Pocket PC 2000 City List Feature
- ActiveX controls Must Implement the IProvideClassinfo and IObjectSafetyinterfaces
- Support All Screen Bit Depths (color and grayscale)
- No File Menu Item
- No User-Exposed Method for Closing the Application

Changed:

- Support for Standard Wait Cursor – a progress bar may be used
- Support for Common Dialog Boxes – Games are not required to use common dialog boxes

Important Notes on International and Localized Versions

Applications must be separately tested in each language in which they are to be distributed. For discounts on multiple tests completed simultaneously, contact the certification test labs (see links on page 3).

You do not need to notify Microsoft if distributing the product internationally unless distribution includes the People's Republic of China (PRC), Taiwan, or Korea. If you are distributing to these countries, please send the Windows Logo Department (winlogo@microsoft.com) an e-mail with your company name as well as the title(s) being distributed and countries where they are being distributed. Please note the following language from the License Agreement:

> "The license right set forth in Section 2(a) shall not extend to the Republic of China ("Taiwan"), South Korea ("Korea"), or the People's Republic of China ("PRC"), unless and until COMPANY provides MS with written notice of COMPANY's intent to distribute Product in these countries. COMPANY agrees not to use the Logo in such countries and shall not be licensed pursuant to this Logo Agreement to do so until COMPANY has provided MS with such written notice."

Important Notes on Application Types

- Applications may be file-based or non-file based. See below for additional requirements for file-based applications.
- Special cases: Development tools, as well as browser plug-ins and add-in products such as graphics, filters, custom dictionaries, or other non-executables that target Windows Mobile 2002-based Smartphones, may earn the logo. Specific notes and requirements for some of these products are included in this document; additional standards will be published as other categories of products emerge. Products that fall into these categories will be handled on a case-by-case basis.

- Hardware, which is bundled with your software, must be tested by the Windows Hardware Quality Lab. E-mail mailto:wcelogo@microsoft.com or visit http://www.microsoft.com/hwdq/hwtest/.

GENERAL REQUIREMENTS FOR SMARTPHONE APPLICATIONS

The following guidelines for Smartphone 2002 applications, which bear the Designed for Windows Mobile logo, were fashioned with the end-user in mind. The goal behind these requirements and recommendations is to foster ease of use by providing a consistent user interface and consistent operation. Like the other Microsoft logo programs, the Designed for Windows Mobile logo is intended to inform consumers that the product bearing the logo complies with a set of criteria that ensures a convenient and predictable user experience.

Installation/Un-installation Requirements

Applications licensed or created by OEMs for distribution exclusively in ROM are exempt from meeting the installation/uninstallation requirements, as well as the cross-platform requirements. However, if the application is distributed by means other than in ROM and requires a setup program, the application must meet all requirements as outlined here.

Required: Use Smartphone Installation Mechanism

Applications must be installed to the Windows Mobile-based Smartphone platform using the Smartphone Application installation mechanism. The Cabwiz SDK setup tool must be used to create CAB files.

Required: Shortcuts in Programs Folder Created on Install

Developers are required to create shortcuts for their applications within the \Windows\Start Menu\Programs folder on the Smartphone device. To ensure the shortcut is in the correct location regardless of device configuration, use the appropriate CE String in the CAB file.

Setup should only create a single shortcut for each application on the device.

Required: Packages Must Have Setup XML (Smartphone WAP Provisioning)

The Setup XML is generated automatically by CABWizSP. If you wish to create your CAB manually, you will also have to create this Setup XML manually.

Required: Provisioning XML File Must Include Install CSP XML with NoUninstall Parm

The provisioning XML file should include an Install CSP section at the start of the document, just below the <wap-provisioningdoc> element. This Install CSP XML should also include the NoUninstall parm.

Required: CESetup DLL Must Expose Correct Four Interfaces

Windows Mobile 2002-based Smartphone supports the optional use of CESetup DLLs. These .dll files are called at various points during an installation or subsequent uninstallation. A CESetup DLL, if included, must have four entry points: Install_Init, Install_Exit, Uninstall_Init, and Uninstall_Exit.

Required: Install CSP Section of Setup XML Must Contain Valid Values for Each of the Parameters

The following requirements must be met for the Install CSP section of the Setup XML file. **Note:** Because CabWizSP will automatically generate this XML correctly, this is only a requirement for CABs that are generated by using any other method.

```
<characteristic type="Install">
  <parm name="InstallPhase" value="install" />
  <parm name="AppName" value="Microsoft Blackjack" />
  <parm name="NumDirs" value="1" />
  <parm name="NumFiles" value="1" />
  <parm name="NumRegKeys" value="0" />
  <parm name="NumRegVals" value="0" />
  <parm name="NumShortcuts" value="0" />
</characteristic>
```

Required: Application Name Must Be Less Than 70 Characters

The application name parameter “AppName” in the Install CSP XML must be less than 70 characters and include the company name followed by the application identifier; for example:
Microsoft Blackjack 1.0.

Required: Application Must Install/Uninstall Correctly

Applications must meet the following criteria to correctly install and uninstall:

- The installation or uninstallation operation must not crash, lock, or otherwise disable any of the functionality of the device.
- All expected confirmation and information dialogs should appear to the user, including installation confirmation, installation progress, removal confirmation, and any dialogs specified by the CESetup DLL.
- The installation and uninstallation logs must report zero errors.

Required: Files Must Be Removed When Uninstalling Application

When the application is uninstalled, any files placed on the device during install or at a later time during execution must be removed (except for shared data files).

Required: Registry Keys Must Be Removed When Uninstalling Application

When the application is uninstalled, any registry keys placed on the device during install or at a later time during execution must be removed (except for shared registry keys).

Required: Applications Must Not Add Files to RAM File System During Installation

During installation, applications must not put any files in RAM file system. Files must be installed in persistent storage (local or removable, i.e.,\StorageCard). To determine acceptable storage, applications can use the *GetStoreInfo* API.

Required: CESetup DLL Must Install to Local Storage Volume

If specified in the installation instructions, the CESetup DLL must correctly install (and run) regardless of its location on any local storage volume (e.g., IPSM, or MMC) if the InstallDir variable is specified and sufficient space exists.

Required: Applications Cannot Be CPF Files

A .cpf file is a provisioning file that is wrapped in a .cab file and can be signed optionally with a certificate. Applications cannot be .cpf files because .cpf files are purely configuration data and do not include any applications.

Required: No CESetup DLL for HME or TSK Files

Home screen files (.hme or .tsk) should contain only home screen related files such as graphics, color schemes, and home screen plugins. CESetup DLLs are not allowed for .hme or .tsk files.

Required: Input Information

When submitting an application, a .cab file is required. Optionally, a Setup package that works on the desktop through Microsoft ActiveSync ® may be submitted.

Required: Clean Up Data from Files and the Registry When Uninstalling

When the user uninstalls, the application should clean up any data from files or the registry, and leave as little behind as possible. The application should use the "Uninstall_Init()" and "Uninstall_Exit()" functionality of a CESetupDLL to clean up data files and databases.

Required: Store DLLs Only in Windows or Application Directory

Any DLL files used in the installation of the application on the Windows Mobile 2002-based Smartphone platform should store only "shared" files or DLLs in the Windows directory, and store every other file in the application's own directory (which may be modified by the end-user). Since the user may change the destination directory on the device during installation, applications can determine their current destination directory on the device during runtime by the registry entry:

HKLM\Software\Apps\<Provider> <AppName>\InstallDir

where <Provider> and <AppName> are the required strings specified in the Setup INF file.

Additionally, if the application installs GX.dll (GAPI), it must install that DLL in the application directory, not the Windows directory. Applications that use GAPI must install this DLL.

UI/Shell Support Requirements

Consistency of the Menu Bar and NavBar is very important to Smartphone applications.

Required: Clean Up When Closing the Application

Applications must be written so that they recognize the request to close the application and perform cleanup. The specific messages that the application will receive include:

- WM_CLOSE (sent when the application is being asked by the Smartphone to shut down completely).

  Recommended: WM_HIBERNATE (sent when memory is running low and the application should clean up any memory that it doesn't absolutely require).

For more information, see the Smartphone SDK.

Required: Menu on Right Soft Key

If the application uses a menu, the menu must be located on the right soft key. The left soft key must be for quick (common) actions, such as "New." If a second menu is used, it can be put on the left.

Required: Dialog Box Controls Must Be Stacked Vertically

If an application includes a dialog box, such as a list of options, all of the controls must be stacked vertically.

Required: No Ellipses for Menu Items

An ellipsis is sometimes used on desktop and other mobile device menus. For Smartphone 2002, menu items cannot include an ellipsis (…).

Required: Use "Done" to Close a Screen

When a screen needs to be closed (for example, when an Option dialog box needs to be closed or when finished adding a contact), "Done" must appear on the left soft key.

Required: Use "Cancel" in Edit View

On a screen where edits can be made, "Cancel" must appear on the right soft key, or in the menu on the right soft key.

Required: Back Button Performs Backspace in an Edit Control

On a screen where text can be edited (for example, in a mail message), the back button must enable the user to backspace on the entire screen, but not exit.

Required: Spinner Controls

If an application requires radio-button or drop-down list behavior, spinner controls (left and right arrows) must be used.

Required: User Must Initiate Call by Choice or Acknowledgement

If an application initiates a voice call, the user must actively choose to make the call or acknowledge a prompt (if the application initiates the call in the background).

Required: No Information Added to Title Bar

Applications cannot add additional icons to the title bar. It is permissible to replace the application title in the title bar with context-specific information. For example, the agenda view in calendar shows the date in the title bar instead of “Calendar”.

Required: 16x16 and 32x32 Pixel Icons for Application and File Types

Applications are required to register 16x16 and 32x32 pixel icons for their main executable and saved file types.

Required: No Duplication of Functionality Provided by Microsoft Pocket Outlook Object Model

Applications that use PIM-type data (appointments, contacts, tasks) must use the Microsoft Pocket Outlook Object Model. This is required to maximize available user storage space by preventing the storage of superfluous PIM data items. Additionally, use of the Microsoft Pocket Outlook Object Model provides a consistent user interface and simplifies communications conducted with a Smartphone.

Required: Microsoft MAPI for the Smartphone Functionality Not Duplicated

Likewise, applications are required to use the Universal Inbox provided by MAPI (Message API) where appropriate. For example, some communication applications (such as two-way paging applications) may require greater functionality than MAPI provides, and are exempt from this requirement. All other applications, however, are expected to use the Universal Inbox provided with Smartphone.

Required: Support for Standard Wait Cursor

Applications must display the system wait cursor (color wheel) when asked to execute a command that renders the current window, or the system as a whole, unresponsive to user input for more than 0.5 seconds. When the system wait cursor is displayed only in the current window, the user may continue to interact with other windows, including the NavBar. The application may present a progress bar instead of the Standard Wait Cursor, if it is loading, storing or rendering data.

Required: Applications Must Respect User Settings

The application must keep the user’s settings, including regional settings, theme colors, and so on. For example, applications must use GetSysColor() to retrieve colors for the UI wherever possible.

Functionality Requirements

Required: Full Functionality and Stability

Applications for Windows Mobile 2002-based Smartphone must be fully functional and stable on the designated test platforms. All functionality must be in place when the application is submitted for testing. While logo testing is not QA testing, the goal is to

confirm that the application being tested does not appear to adversely affect the overall stability or performance of the Smartphone 2002 device environment. The application must recover from standby/suspend situations on leading equipment. The application must also be well behaved with Shell and system features, such as not compromising hardware button functionality or overriding other Shell or system features, unless it is required for the proper functioning of the application.

Test platforms for the test will be determined by the operating system requirements specified for the application.

Required: Must Not Assume External Storage

Although many Smartphone devices have them, external storage cards are not required on a Smartphone device. Applications must not crash, hang, or exhibit adverse behavior on devices without external storage cards.

Required: Graceful Application Shutdown

Because applications will be shut down by the Shell, without the user taking any explicit action, applications must shut down gracefully. This means not displaying any dialog boxes or other UI; not consuming excessive CPU time; and not consuming significant additional memory while closing.

Required: Restoration of State When Starting

Because the Smartphone user will not have any concept of which applications are and are not running, applications must restore critical state when launched. Note that this requirement does not mean that complete state must be restored, although this would be ideal. The goal is for the user to feel comfortable returning to the application, whether or not it was shut down since last used. For example, applications should restore the user to the previously selected view and scroll state, if applicable.

Required: No Accelerators Shown for Menu Items

When there is no physical keyboard on the Smartphone, no letters should be used as accelerators, and numbers should not be put in the front of the items to indicate their associated numeric keypad association.

Recommended: Avoid manually assigning mnemonics to menu items or dialog boxes.

Required: Applications Must Use the Connection Manager to Configure All Connectivity Options

Connection Manager provides the user with a single user interface from which they can configure all their connectivity options – wired and wireless network cards, modems, cellular, VPN, and so on. It also exposes a simple API, allowing the developer to essentially ask Connection Manager to provide an Internet or “work network” connection, and leaves the system to sort out the details.

Required: Long Filename Support

The application must:

- Support long filenames as described below.

- Use long filenames for displaying all document and data filenames in the Shell, in dialogs and controls, and with icons.
- It is strongly recommended that the .XXX extensions are hidden in the application itself.

Smartphone applications that save files that are exposed to the user must support long filenames (LFNs) with all the following required features:

- Users must be able to enter filenames of 128 characters, including all uppercase and lowercase standard characters, embedded spaces, embedded periods, and so on.
- Leading and trailing spaces must be stripped by the Save As command.
- It is not necessary to allow leading and trailing periods. These may be stripped by your application if you wish.
- Question marks anywhere in the filename must prevent the file from being saved. No error message needs to be displayed.
- The plus-sign (+), comma (,), semicolon (;), equals-sign (=), left-square bracket ([), and right-square bracket (]) must be supported anywhere in the filename, including leading and trailing. Embedded periods must also be supported. These should not cause any error conditions.

Required: Shut Timers Off When Application Is Running in the Background

To optimize battery use on the Smartphone device, if an application uses a timer for visual elements, the timers must shut off when the application is no longer running in the foreground.

Required: Applications Must Not Interfere with Incoming Call Functionality

An application cannot interfere with the normal call user interface, and it cannot delay the user's ability to answer an incoming call. Applications that run in full screen mode or always on top must move to the background to allow the incoming call user interface to appear on top.

SPECIAL CIRCUMSTANCES AND ADDITIONAL REQUIREMENTS

SDI/File-Based Applications

These are applications with the primary purpose of opening, creating, and editing documents. Word processors, spreadsheets, and so on, are all considered file-based applications. (Examples of non-file-based applications include PIMs and games.) File-based applications are subject to the following additional requirements.

Required: Support Only One Instance of Each Application or Control Panel

Because there is no taskbar and memory is managed by the Shell, users switch between applications that are still running using the Start menu or Programs folder. As a result, only one instance of each application or applet must be allowed to run. Any application that supports multiple open documents or data types must support that functionality from within the application, not through multiple instances.

Utilities

Utility products are non-file-based applications or application components designed to optimize the Smartphone environment. Some requirements may not apply to certain utilities, and will be considered on a case-by-case basis.

===Related Information===
An example of a utility is bBackup by BSQUARE. It backs-up user's data.

Add-ons

Add-on products are generally non-executables such as data libraries, clip art collections, templates, and macros. Add-ons must operate with an application that bears the logo and must fulfill all relevant logo requirements for installation, and so on.

Device Drivers

Device Drivers must fulfill all relevant requirements for UI and Windows CE .NET Test Kit (CETK) where applicable.

Hardware

Compatible hardware peripherals may qualify for a *Designed for Windows Mobile* logo and participate in the Mobile2Market program if they work as advertised for specific Smartphone models and can be classified within the following groups:

- Hardware accessory
  - CF/SD Memory, stylus, case, etc.
- Hardware – proprietary connection without software
  - Head set, Car Kit, charger, cable, etc.
- Hardware with device driver
  - GPS, keyboard, network, etc.
- Hardware with driver and UI
  - Keyboard, network card, GPS, barcode scanner, camera, etc.

The certification process requires the hardware and device driver, if needed, be submitted to NSTL for testing and Microsoft Windows CE.Net Test Kit (CETK) review. Specific details may be obtained from NSTL (http://www.nstl.com/logoprogram/win_ce_logoprograms.html).

OTHER MICROSOFT LOGO PROGRAMS

Microsoft offers a number of logo programs, which all share a common goal: providing the end user with an easy way to recognize products that were designed to work well with Microsoft's industry-standard operating systems and applications. Customers of products that bear the logo can more easily find products compatible with their existing systems and take advantage of the functional skills and knowledge they already possess. It is possible for applications to bear more than one logo, and developers are encouraged to take advantage of the full array of marketing opportunities presented by these programs.

Microsoft Certification Programs:

- Certified for Windows 2000 (http://msdn.microsoft.com/certification/)
- Designed for Microsoft Windows XP (http://www.microsoft.com/winlogo/default.asp)
- Microsoft Commerce Server 2000 Integration Testing (http://www.microsoft.com/commerceserver/partners/default.asp)
- Windows 2000 Independently Verified Compatible (http://www.microsoft.com/Partner/Partnering/certified/default.asp)
- Featuring Microsoft Visual Basic Technology (http://msdn.microsoft.com/vba/license/process.asp)
- Microsoft Windows Terminal Services Testing (http://www.microsoft.com/ntserver/ProductInfo/terminal/default.asp)